SISTEMAS DE NUMERACIÓN

Sistema decimal

Desde antiguo el Hombre ha ideado sistemas para numerar objetos, algunos sistemas primitivos han llegado hasta nuestros días, tal es el caso de los "números romanos", pero sin duda el más extendido en la actualidad es el sistema decimal de números arábigos, llamado así por ser los árabes sus creadores.

En el sistema decimal, los números se forman por combinación de 10 signos distintos; 0, 1, 2, 3, 4, 5, 6, 7, 8 y 9. Cada uno de estos signos tiene un valor, y el valor del número que forman se halla multiplicando el valor de cada uno de ellos por 10 elevado a la potencia correspondiente a su posición en el número, siendo 0 el de más a la derecha, 1 el siguiente y así sucesivamente. De esta forma, el número 5348 sería igual a:

5348 = 8*10^0 + 4*10^1 + 3*10^2 + 5*10^3 = 8*1 + 4*10 + 3*100 + 5*1000

La misma denominación del número nos lo recuerda, decimos: cinco mil, tres cientos, cuarenta y ocho. El sistema decimal es de uso tan frecuente que no vale la pena insistir en él, pero es importante hacer notar que la base de los exponentes es siempre 10 y por tanto, este sistema se denomina también "de base 10". Es posible crear sistemas que utilicen una base distinta, y de hecho, estos sistemas son muy usados en informática.

Sistema binario

Un ordenador es una máquina esencialmente binaria, su componente básico es el transistor, que sólo admite dos estados posibles, o bien pasa corriente, o bien no pasa.

Algunos podrían objetar que un transistor puede tener múltiples estados dependiendo de la cantidad de corriente que pase; es cierto, pero la medición de esta cantidad de corriente implica una imprecisión que podría crear ambigüedades, y un ordenador no admite ambigüedad. De forma que por debajo de un determinado valor, se considera que no pasa corriente, y por encima de otro, se considera que sí pasa.

Arbitrariamente, asociamos estos dos estados con dos dígitos, cuando no pasa corriente decimos que tenemos un "0" y cuando pasa, decimos que tenemos un "1". De esta forma, podremos representar el estado de un interruptor: "1" cuando esté encendido y "0" cuando esté apagado.

Si tenemos una serie de interruptores puestos en fila, podríamos representar el estado de todos ellos mediante un número binario.

En la figura siguiente vemos una serie de interruptores cuyo estado podría ser definido mediante el número binario: "10011".

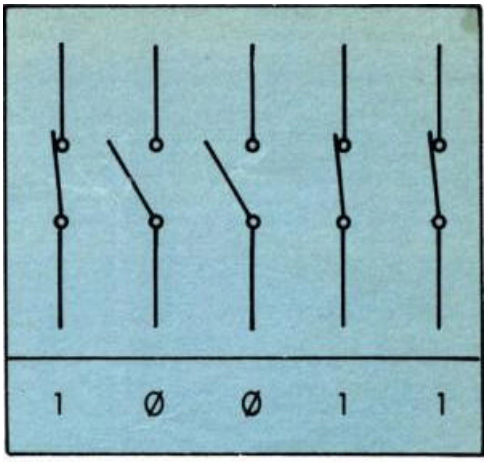

Figura 1.

Como se ve, el sistema binario es perfectamente adecuado para su uso en circuitos electrónicos. A cada "1" o "0" de un número binario le llamaremos "dígito binario", que puede abreviarse como "bit" (contracción de "binary digit").

El valor de un número binario se halla de la misma forma que en el sistema decimal, excepto que esta vez, la base es "2". Así el número "10011" será:

10011 = 1*2^0 + 1*2^1 + 0*2^2 + 0*2^3 + 1*2^4

Es decir:

10011 = 1*1 + 1*2 + 0*4 + 0*8 + 1*16 = 19 en decimal

Ya hemos visto implícitamente, cómo transformar un número binario en decimal, el proceso inverso (transformar un número decimal en binario), lo veremos más adelante, cuando estudiemos el método general para transformar números en cualquier base.

Operaciones aritméticas en binario

Los números binarios se pueden sumar, restar, multiplicar y dividir de igual forma que los decimales, sólo es necesario conocer las "tablas" correspondientes. Veamos primero la suma.

Para sumar en decimal los números 19 y 28, los colocamos de la siguiente forma:

```
+ 19
  28
```

Y a continuación hacemos: "9 más 8 igual 7 y me llevo 1, 1 más 1 más 2 igual 4" el resultado es 47, es decir:

```
+ 1
  19
  28
= 47
```

Al sumar 9 y 8 nos da un resultado superior a 9, es decir, superior al dígito más alto de nuestro sistema, por lo que decimos que "nos llevamos una", esta "una" que "nos llevamos" se denomina en binario "acarreo" (o "carry" en inglés). Cuando se suma en binario, es necesario tener en cuenta el acarreo.

Vamos ahora a ver la "tabla de sumar" en binario:

```
0 + 0 = 0
0 + 1 = 1
1 + 1 = 10
```

Es decir, cuando sumamos 1 más 1 el resultado es 0 pero se produce un acarreo de 1.

Veamos un ejemplo: vamos a sumar los números 19 y 28 en binario, 19 es 10011 y 28 es 11100, de esta forma:

```
+ 1
  010011
  011100
= 101111
```

El resultado es 101111, o bien 47 en decimal. Al sumar los dos unos de la izquierda, se ha producido un acarreo que hemos anotado encima de la siguiente columna. Al igual que en decimal, los ceros a la izquierda no tienen valor.

Números negativos

Hemos visto cómo sumar, pero ¿cómo se resta? Para conseguir restar en binario, tendremos que establecer antes un convenio sobre qué consideramos números negativos.

Se denomina "complemento a 1" de un bit a lo que le falta a ese bit para ser "1", es decir, el complemento de "1" es "0" y el de "0" es "1" . Por tanto, el complemento a 1 de un número binario es el resultado de cambiar sus "unos" por "ceros" y sus "ceros" por "unos".

Por ejemplo: el complemento a 1 de "10011101" es "01100010".

Por otro lado, se denomina "complemento a 2" al "complemento a 1" más 1, es decir, al resultado de cambiar los unos por ceros y los ceros por unos, y luego sumar uno. Veamos algunos ejemplos:

Podría parecer algo arbitrario, sin embargo es sumamente útil, ya que como veremos a continuación, es posible usar el complemento a 2 de un número como su negativo. Para restar dos números, sumamos al "minuendo" el complemento a 2 del "sustrayendo".

Vamos a ver algunos ejemplos, trabajando con números de 8 bits. Vamos a restar 28 menos 19, para lo cual sumamos a 28 el complemento a 2 de 19:

```
   - 28
     19
   = 9

  + 00011100
    11101101
= 100001001
```

Obtenemos el número "00001001" con un acarreo de "1". El acarreo nos indica que el resultado es positivo, es decir, el resultado es "+9" como era de esperar.

Ahora vamos a realizar la operación inversa, es decir, vamos a restar 19 menos 28 por tanto, tendremos que sumar a 19 el complemento a 2 de 28:

```
    - 19
      28
    = -9

   + 00010011
     11100100
  = 011110111
```

Esta vez hemos obtenido el número "11110111" con un acarreo de "0". El hecho de que el acarreo sea "0" nos indica que el número es negativo, y "11110111" es, precisamente, complemento a 2 de "9", es decir, "-9" como también era de esperar.

Un hábil lector habrá comprobado que, trabajando con números de 8 bits, podemos saber si un número es negativo con sólo mirar el primer bit (el de más a la izquierda); si este bit es "1", el número será negativo. Bien, esto no siempre es cierto. Trabajando con 8 bits, se pueden representar 256 números distintos (desde 0 hasta 255).

Sistema hexadecimal

Como hemos visto, los números binarios son muy adecuados para su uso en aparatos electrónicos, pero tienen un gran inconveniente; cuando los escribimos, necesitamos una gran cantidad de cifras para representar un número relativamente pequeño. Si pudiéramos agrupar los bits, conseguiríamos evitar este inconveniente.

Dado que vamos a trabajar con 8 o 16 bits, parece lógico agruparlos de 4 en 4 con el fin de obtener números de 2 o 4 cifras. Como regla general, con "n" bits se pueden obtener "2 elevado a n" combinaciones distintas, por tanto, con 4 bits podemos obtener 16 combinaciones, cada una de las cuales las asociaremos con un dígito hexadecimal.

Necesitamos 16 dígitos para representar todas las posibles combinaciones, como sólo conocemos 10 dígitos distintos (del 0 al 9), utilizaremos las 6 primeras letras mayúsculas del abecedario (de la "A" a la "F").

En la figura siguiente se pueden ver las 16 combinaciones posibles con 4 bits y su equivalente en hexadecimal.

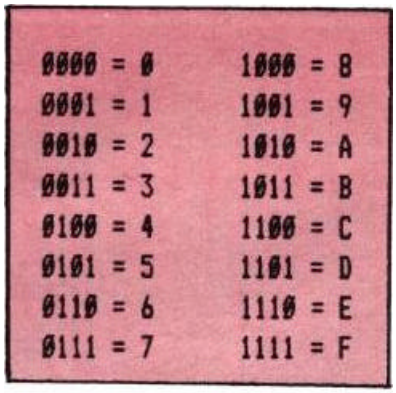

Figura 3.

Supongamos el número binario "01101100", siguiendo la tabla de la figura, podemos escribirlo como "6Ch". Hemos escrito "6" en lugar de "0110" y "C" en lugar de "1100", la "h" se añade al final para indicar que se trata de un número hexadecimal y no confundirlo con uno decimal. A los números hexadecimales se les denomina con frecuencia, simplemente, "Hexa".

La forma de transformar un número Hexa en decimal, es sumamente sencilla, basta con multiplicar el valor de cada dígito por 16 elevado a la correspondiente potencia (como hacíamos anteriormente para los binarios y decimales); habrá que tener en cuenta, que "A" vale 10, "B" vale 11, "C" vale 12, "D" vale 13, "E" vale 14 y "F" vale 15. Veamos algún ejemplo, vamos a convertir a decimal el número "5CB2h":

5CB2h = 2*16^0 + B*16^1 + C*16^2 + 5*16^3

es decir:

5CB2h = 2*1 + 11*16 + 12*256 + 5*4096 = 23730

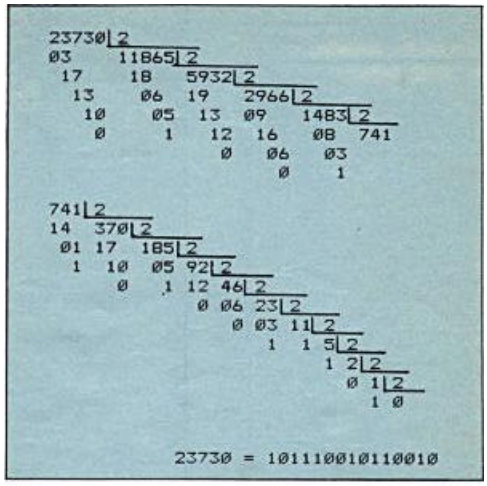

Figura 5.

CAMBIOS DE BASES :

CONVERSION DE BINARIO A DECIMAL

Cualquier numero binario puede convertirse a su equivalente decimal, simplemente sumando en el numero binario las diversas posiciones que contengan un 1.

EJEMPLO

1 1 0 1 1 (BINARIO)

$2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 16 + 8 + 2 + 1$

= 27 (DECIMAL)

CONVERSION DE DECIMAL A BINARIO

Existen 2 maneras de convertir un número decimal entero a su representación equivalente en el sistema binario. En el primer método el número decimal se expresa simplemente como una suma de potencias de 2 y luego los unos y los ceros se escriben en las posiciones adecuadas de los bits.

EJEMPLO

$45_{10} = 32 + 8 + 4 + 1 = 2^5 + 0 + 2^3 + 2^2 + 0 + 2^0$

$= 1\ 0\ 1\ 1\ 0\ 1_{\ 2}$

$76_{10} = 64 + 8 + 4 = 2^6 + 0 + 0 + 2^3 + 2^2 + 0 + 0$

$= 1\ 0\ 0\ 1\ 1\ 0\ 0_2$

En el segundo método se emplea la división repetida por 2. La conversión requiere dividir repetidamente el número decimal entre 2 y que se escriban los restos después de cada división hasta que se obtenga un cociente de 0.

EJEMPLO

25 =12 + RESTO 1
2
12 = 6 + RESTO 0
2
6 = 3 + RESTO 0
2
3 = 1 + RESTO 1
2
1 = 0 + RESTO 1

$25_{10} = 1\ 1\ 0\ 0\ 1_2$

CONVERSION DE DECIMAL A OCTAL

Un entero decimal se puede convertir a octal con el mismo método de división repetida que se usó en la conversión de decimal a binario, pero con factor de división de 8.

EJEMPLO

266 = 33 + RESTO DE 2
8
33 = 4 + RESTO DE 1
8
4 = 0 + RESTO DE 4
8
$266_{10} = 4\ 1\ 2_8$

CONVERSION DE OCTAL A BINARIO

La ventaja principal del sistema de numeración octal es la facilidad con que se puede realizar a conversión entre números binarios y octales. La conversión de octal a binario se lleva a cabo convirtiendo cada dígito octal en su equivalente binario de 3 bits. Los ocho dígitos posibles se convierten como se indica en la siguiente tabla.

Dígito Octal	0	1	2	3	4	5	6	7
Equivalente Binario	000	001	010	011	100	101	110	111

Por medio de estas conversiones, cualquier número octal se convierte a binario, convirtiéndolo de manera individual. Por ejemplo, podemos convertir 472_8 a binario de la siguiente manera:

4 7 2
100 111 010

por tanto el número octal 472_8 es equivalente a binario **100111010_2**.

5431_8

101 100 011 001

CONVERSION DE BINARIO A OCTAL

La conversión de enteros binarios a octales es simplemente la operación inversa del proceso anterior. Los bits del número binario se agrupan en conjuntos de tres comenzando por la derecha. Luego, cada grupo se convierte a su equivalente octal.

EJEMPLO

1 0 0 1 1 1 0 1 0
$4\ 7\ 2_8$

Algunas veces el numero binario no tendrá grupos pares de 3 bits. En estos casos, podemos agregar uno o dos ceros a la izquierda del numero binario a fin de completar el ultimo grupo. Esto se ilustra a continuación para el numero binario 11010110.

0 1 1 0 1 0 1 1 0
$3\ 2\ 6_8$

Nótese que se agregó un cero a la izquierda para producir grupos pares de tres bits.

CONVESION DE HEXADECIMAL A DECIMAL

El sistema hexadecimal emplea la base 16. Así tiene 16 posibles símbolos digitales. Utiliza los dígitos del 0 al 9 más las letras A,B,C,D,E,y F como sus 16 símbolos digitales. La sig. tabla nos muestra las relaciones entre los sistemas hexadecimal, decimal y binario. Cada dígito hexadecimal representa un grupo de cuatro digitos binarios.
Un número hexadecimal se puede convertir a su equivalente decimal utilizando el hecho de que cada posicion de los dígitos hexadecimales tiene un valor que es una potencia de 16 de manera que si tenemos 3 digitos hexadecimales su primer de derecha a izquierda tomara la potencia de 0 y despues 1, 2 y asi sucesivamente, despues de elevar a la potencia el número hexadecimal lo multiplicara por el digito que le corresponda ejemplo:

$$356_{16} = 3 * 16^2 + 5 * 16^1 + 6 * 16^0$$
$$= 768 + 80 + 6$$
$$= 854_{10}$$

HEXADECIMAL	DECIMAL	BINARIO
0	0	0000
1	1	0001
2	2	0010
3	3	0011

4	**4**	**0100**
5	**5**	**0101**
6	**6**	**0110**
7	**7**	**0111**
8	**8**	**1000**
9	**9**	**1001**
A	**10**	**1010**
B	**11**	**1011**
C	**12**	**1100**
D	**13**	**1101**
E	**14**	**1110**
F	**15**	**1111**